

AD-R141 269

PERFORMANCE ESTIMATE OF A REFLECTING LIGHT PIPE FOR  
INFRARED DETECTORS (U) ROYAL SIGNALS AND RADAR  
ESTABLISHMENT MALVERN (ENGLAND) J WARNER FEB 84

1/1

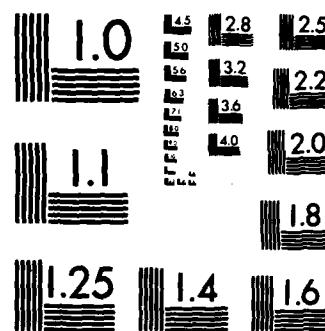
UNCLASSIFIED

RSRE-MEMO-3661 DRIC-BR-91490

F/G 17/5

NL

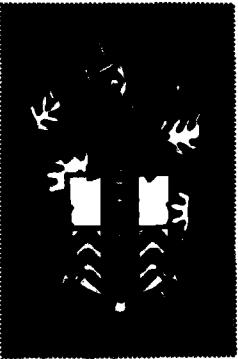




BR91490

UNLIMITED

(2)



RSRE -MEMO-

MEMORANDUM No. 3661

# ROYAL SIGNALS & RADAR ESTABLISHMENT

AD-A141 269

PERFORMANCE ESTIMATE OF A REFLECTING  
LIGHT PIPE FOR INFRARED DETECTORS

Author: J Warner

RSRE MEMORANDUM No. 3661

DTIC FILE COPY

PROCUREMENT EXECUTIVE,  
MINISTRY OF DEFENCE,  
RSRE MALVERN,  
WORCS.

DTIC  
ELECTED  
MAY 18 1984  
S E D  
E

UNLIMITED

ROYAL SIGNALS AND RADAR ESTABLISHMENT

Memorandum 3661

Title: PERFORMANCE ESTIMATE OF A REFLECTING LIGHT PIPE FOR INFRARED DETECTORS

Author: Dr J Warner

Date: February 1984

SUMMARY

A hollow reflecting cone or pyramid can be used to concentrate infrared radiation on to a small photo-detector. This note describes calculations which predict that a five fold increase in flux density should be readily obtainable.

Accession For	
NTIS	GRA&I
ERIC TAB	<input type="checkbox"/>
Announced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Avail and/or	
Dist	Special
A-1	



Copyright  
C  
Controller HMSO London  
1984

RSRE MEMORANDUM NO 3661

PERFORMANCE ESTIMATE OF A REFLECTING LIGHT PIPE FOR INFRARED DETECTORS

Dr J Warner

CONTENTS

- 1 SUMMARY
- 2 BACKGROUND
- 3 METHOD OF CALCULATION
  - 3.1 RAY-TRACING
  - 3.2 TRANSMITTANCE CALCULATION
  - 3.3 FORMULAE FOR FLUX DENSITY IMPROVEMENT FACTOR
- 4 DISCUSSION OF RESULTS
- 5 REFERENCES
- 6 FIGURES
  - FIG 1 RAY DIAGRAM SHOWING REFLECTIONS INSIDE A PYRAMID REFLECTOR
  - FIG 2 FRACTIONAL TRANSMITTANCE VS. RAY ANGLE
  - FIG 3 NETT TRANSMITTANCE INTEGRATED OVER DIFFERENT RAY-CONES
  - FIG 4 NETT FLUX CONCENTRATION VS. F-NUMBER
  - FIG 5 NETT FLUX CONCENTRATION VS. TRUNCATION RATIO
- 7 APPENDIX. PROGRAM LISTINGS.

## 1 SUMMARY

The RSRE Scanner Optics Ray-tracing Programs have been used to calculate the transmittance of a truncated pyramid reflector that would increase the apparent size of an infrared detector if placed immediately over it. The dependence of flux concentration with pyramid geometry and ray angles is presented graphically and it is shown that concentration factors greater than five should be easy to achieve.

## 2 BACKGROUND

The use of a hollow cone or pyramid with reflecting walls to concentrate radiation on to a small area detector is well known (ref.1). Figure 1. shows a parallel bundle of 30 rays incident at an angle of 10 degrees on to a 3:1 truncated pyramid. Only 8 rays are reflected back out of the pyramid and so the nett ray concentration factor is  $(22/30) \times 3 = 2.2:1$  (rays/unit length).

In the quest for thermoelectrically cooled infrared detectors it has been determined that a 2 or 3 fold increase in flux density at the detector could dramatically improve the chances of successful operation (ref.2). An array of pyramid reflectors could be made to overlay a detector array whose inter-element spacings were increased to match the desired concentration ratio.

This memo reports how the transmittance of a truncated hollow pyramid was determined by ray-tracing methods for several pyramid geometries and ray angles. A simple model was used to describe the ray angle distribution at the focus of a lens and numerical integrations were carried out to estimate the flux concentration factors at the chosen truncation ratios, and pyramid side angles for several lens F-numbers. Graphs are presented (Figs. 4,5) which show how the concentration factor is affected by all these parameters.

## 3 METHOD

### 3.1 THE RAY-TRACING

A square pyramid of  $1/2$  angle  $\theta$  lying along the  $z$ -axis with its apex at the origin is bounded by the four planes

$$z \pm x/\tan(\theta) = 0 \text{ and } z \pm y/\tan(\theta) = 0. \dots \quad (1)$$

If a ray with direction cosines  $l, m, n$  passes through a point  $(a, b, c)$  then another point on the ray, a distance,  $d$ , from  $(a, b, c)$  is at  $(x, y, z)$  where

$$x = a + d*l \quad y = b + d*m \quad z = c + d*n \dots \quad (2)$$

Equations (1) and (2) can be combined to give the distances along the ray from  $(a, b, c)$  to the four planes that form the pyramid;

$$d = -(c*\tan(\theta) \pm b)/(l*\tan(\theta) \pm m) \dots \quad (3)$$

The HP9825 computer subprogram "PYR" listed in the Appendix was used in the RSRE Ray-trace Programs (Ref.3) to locate the point of intersection on the appropriate face of the pyramid. A control flag was set if the intersection point was closer to the apex than that appropriate to the chosen truncation ratio. A different control flag was set if the ray direction had a component in the  $+z$  direction, indicating that the ray had failed to pass through the pyramid and was travelling away from the detector.

### 3.2 TRANSMITTANCE CALCULATION

The procedure "PYR" outlined above can be used to find out if a given ray is transmitted or reflected by a truncated square reflecting pyramid. A ray incident on the open base of the pyramid at a given angle can be scanned in (say) the y-direction to locate the reflection/transmission boundaries for rays of that angle and x-position. Once these y-values have been located for several x-positions a numerical integration procedure will determine what proportion of a parallel ray-bundle filling the pyramid base will pass through to the detector. This fraction, F, was calculated by the program "RT2#6" listed in the appendix for ray angles from 0 to 30 degrees, for pyramid angles from 10 to 25 degrees, and for truncation ratios from 2:1 to 3.5:1. The results are plotted as F vs. Ray angle in Fig.2(A-D).

### 3.3 FORMULAE FOR FLUX DENSITY IMPROVEMENT FACTOR

Let a circular lens aperture be uniformly illuminated. After refraction through the lens the angle between a ray and the lens axis is roughly proportional to the ray height at the lens aperture. The number of rays at a given angle is therefore approximately proportional to the ray angle. If the maximum ray angle is  $a_0$  then the following equations can be written.

$$\text{Total ray flux on lens} = \int_{-\infty}^{\infty} \text{ada} = K^* a_0^2 \quad \dots \dots \dots \quad (5)$$

Dividing eq(6) by (5) will determine the pyramid transmittance for a focussed beam of given numerical aperture ( $NA = \sin(\alpha_0)$ ).

If all radiation incident on the base of a pyramid with (linear) truncation ratio  $R$  passes out of the apex end then the flux density would be increased by  $R^2$ . Taking equation (7) into account the nett flux density increase due to a hollow pyramid concentrator is therefore given by

$$\text{Flux Density Concentration} = \left( \frac{2 \cdot R^2}{a_0^2} \right) \int_{a_0}^{a_0} a f(a) da \quad \dots \dots \quad (8)$$

## 4 DISCUSSION OF RESULTS

Equations (7) and (8) have been evaluated by numerical integration and the results are shown graphically in Fig.3 and Fig.4. In these figures the pyramid geometry is characterised by prism angle and truncation ratio. An alternative parameter is the pyramid height, expressed in units of detector size. Different values of angle and truncation ratio can combine to give the pyramid height ( $h$ ) since  $h = (R-1)/(2\tan(\theta))$ . Flux density concentrations factors have been extracted from fig.4 and re-plotted against truncation ratio for certain F-numbers and pyramid heights. Figure 5 clearly shows that concentration factors in excess of 5 should be obtainable with reasonable pyramid geometries.

It is implied in eq.(6) that  $F(\alpha)$  depends only on the ray incidence angle and not on its azimuth (eg. horizontal, vertical, or diagonal). A square pyramid of

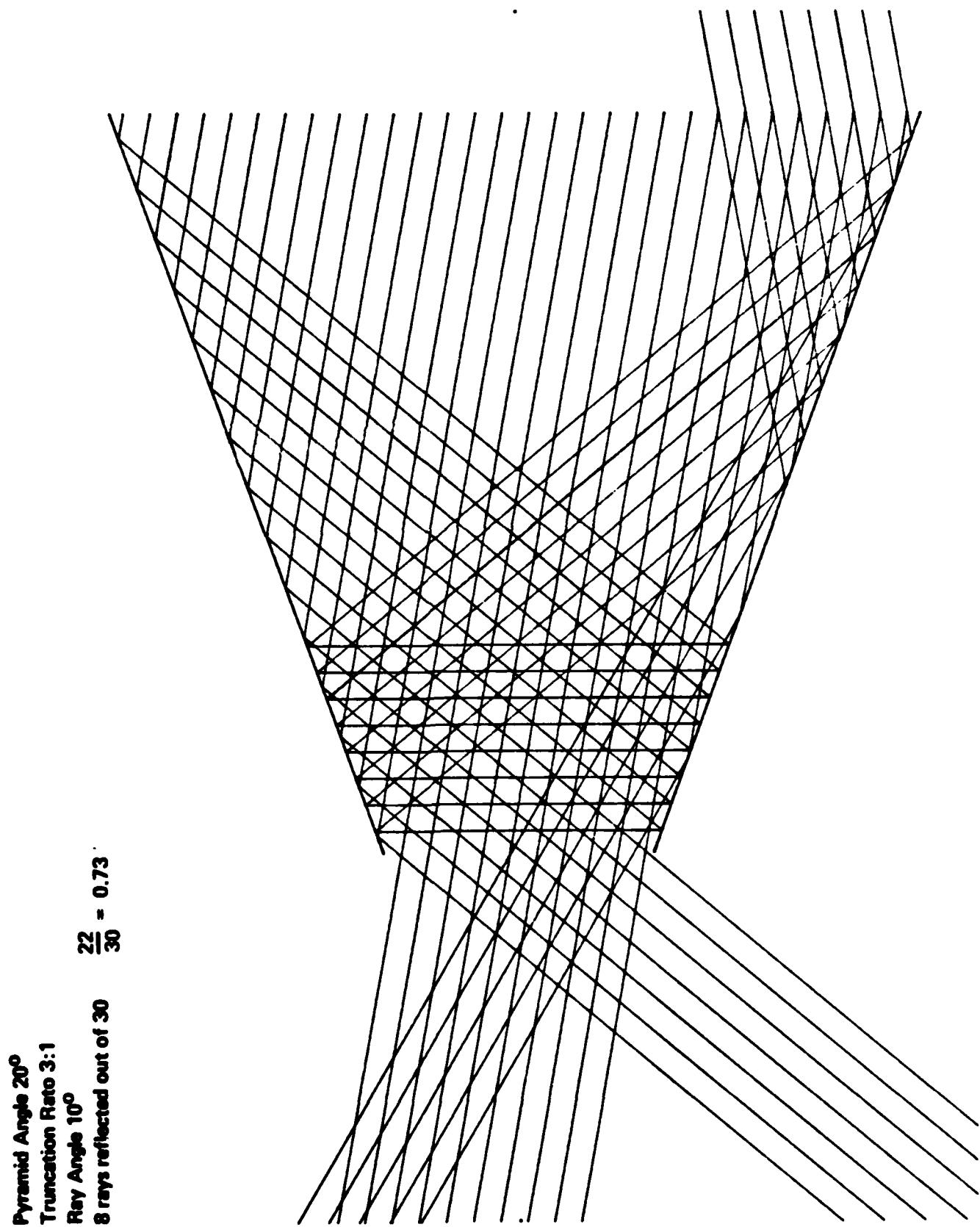
course has 4-fold symmetry about its axis and so the integration in eq.(6) is not strictly accurate. However the variation of  $F(\alpha)$  with azimuth for a given ray angle is expected to be small, making eq.(6) and the others derived from it, good approximations.

A ray-tracing model could be established for a particular lens, pyramid, and detector arrangement to confirm the expected flux density improvements. Such a computation would launch a large bundle of rays through the lens representing uniform illumination (in position and direction) and counting those that pass into and out of the pyramid concentrator. This was done for an F-1.5 Germanium doublet lens feeding a 25 micron detector via a 2.5:1 pyramid reflector 100 microns long. The incident ray bundle comprised nearly 50,000 rays (the ray bundle was intentionally made to spread well beyond the base of the pyramid). 62% of the rays passed into the pyramid, and 93.1% of these reached the detector plane. This implies a flux density concentration of  $.931 \times (2.5)^2 = 5.8$ , within 4% of the estimate of 5.6 from fig.5B.

## 5 REFERENCES

- 1 W. J. SMITH  
Modern Optical Engineering  
McGraw Hill (1966) p234
- 2 D. J. LEES & R. A. BALLINGALL  
RSRE Memo (to be published)
- 3 J. WARNER  
Ray-tracing Technique for Infrared Scanners  
Proc IEE Conf on Low Light & Thermal Imaging  
IEE Conf pub.173 (1979) p20

FIG 1



$F(a)$  vs  $a$   
For different pyramid angles

FIG 2A

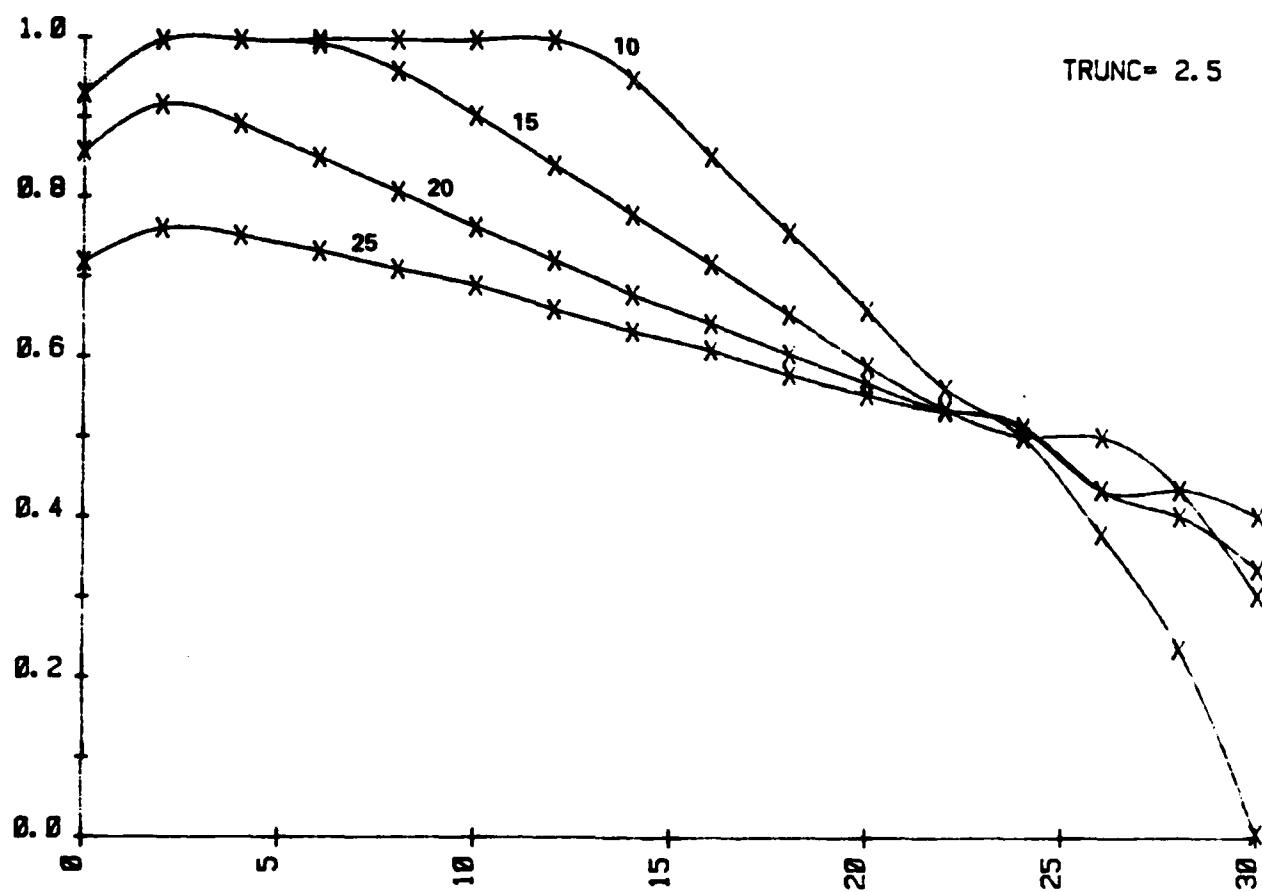
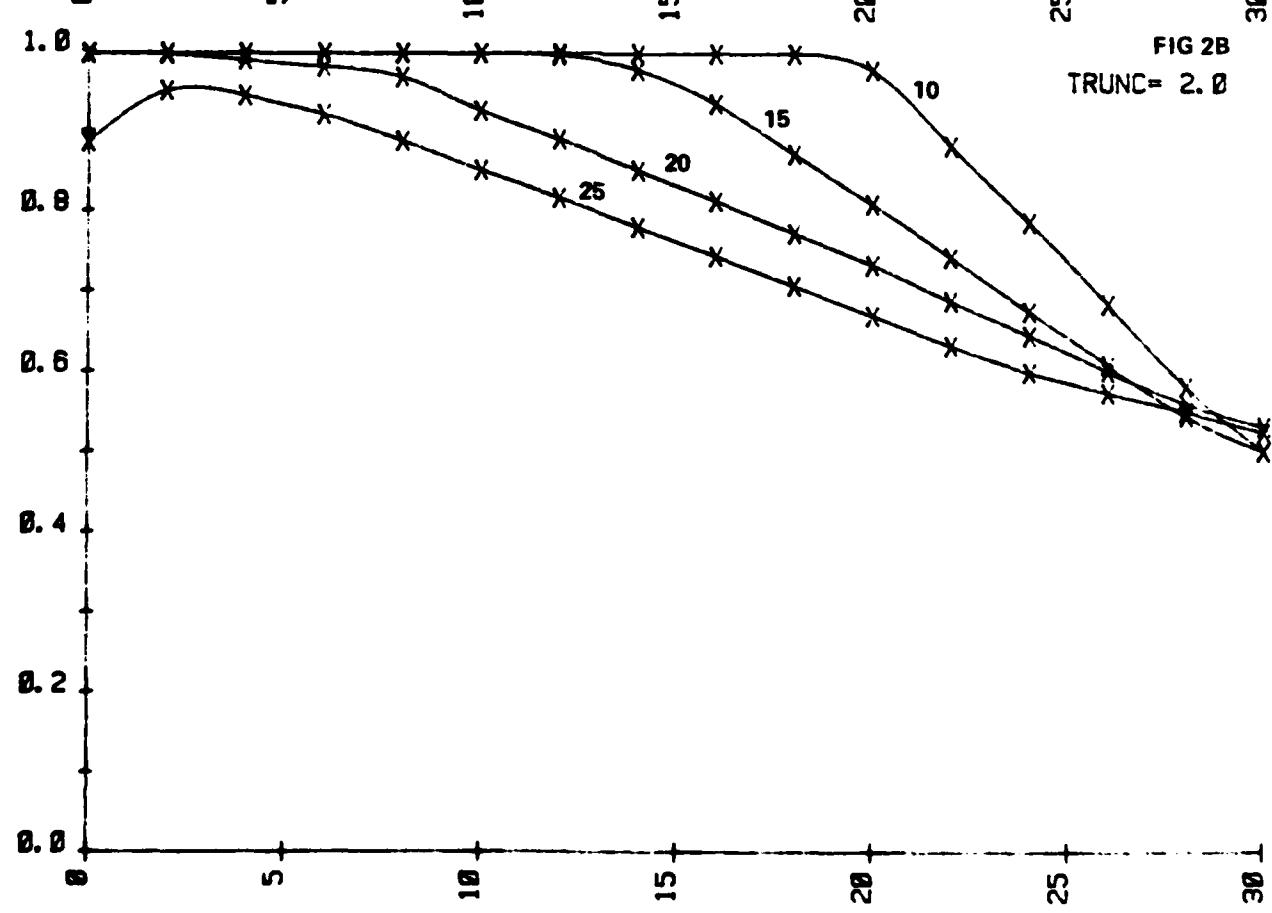


FIG 2B  
TRUNC = 2.0



$F(\alpha)$  vs  $\alpha$   
For different pyramid angles

FIG 2C

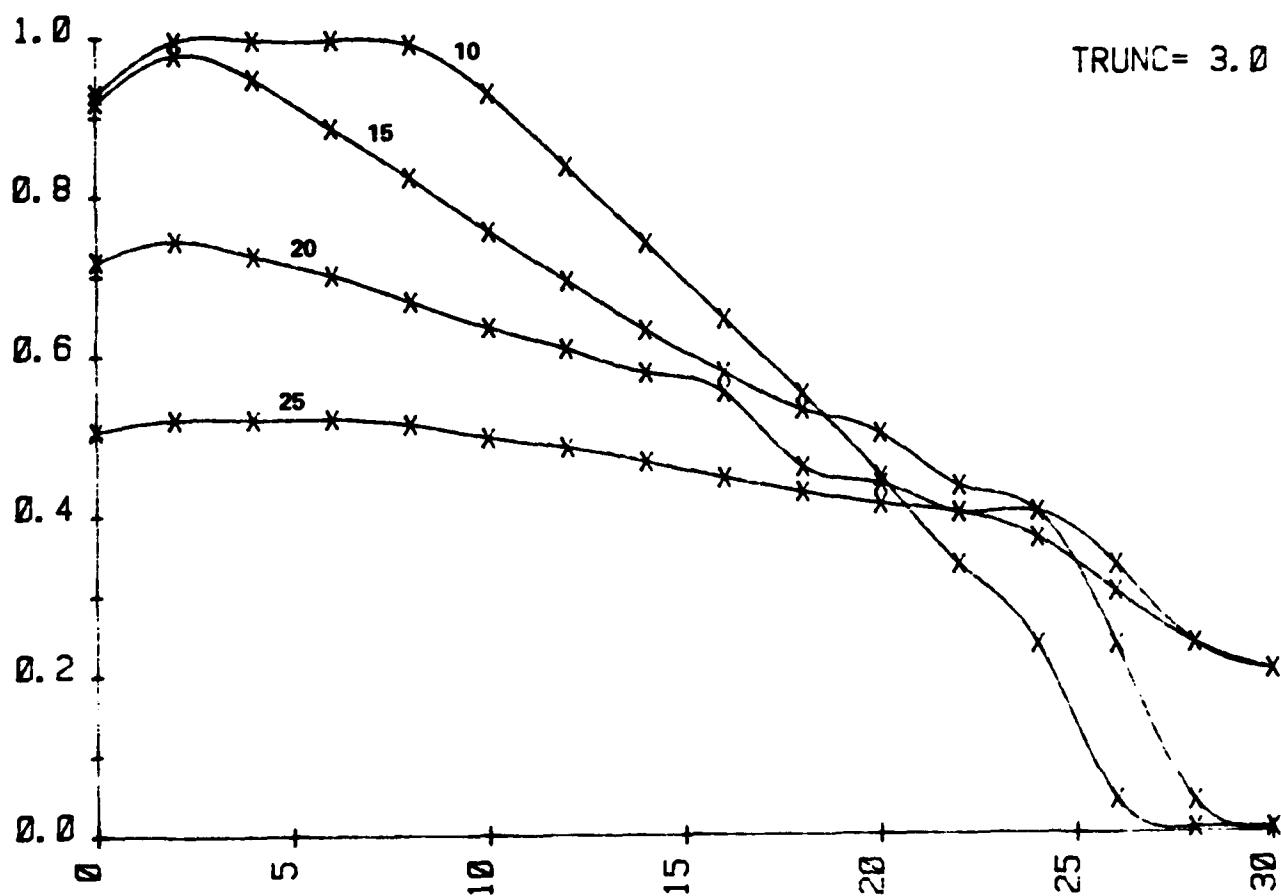


FIG 2D

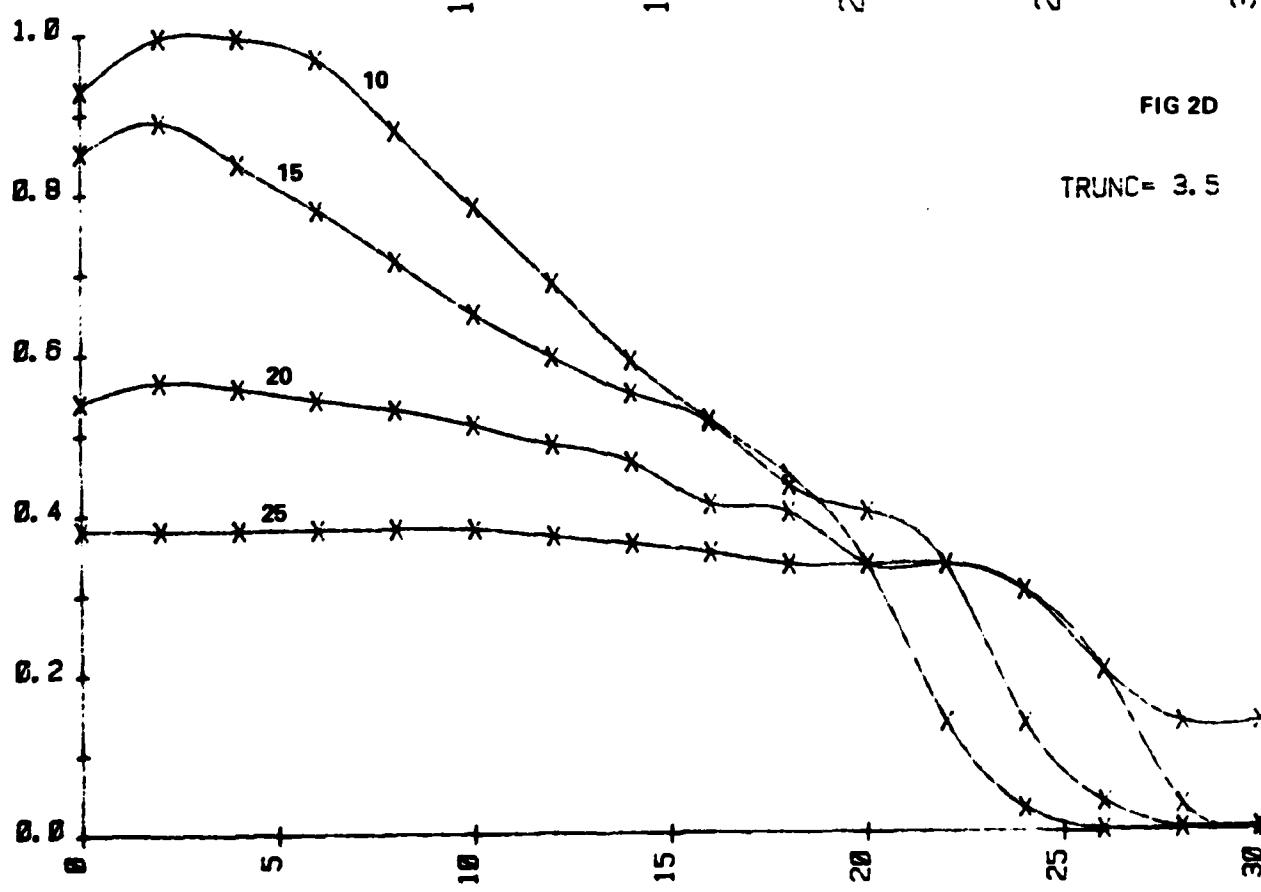


FIG 3A

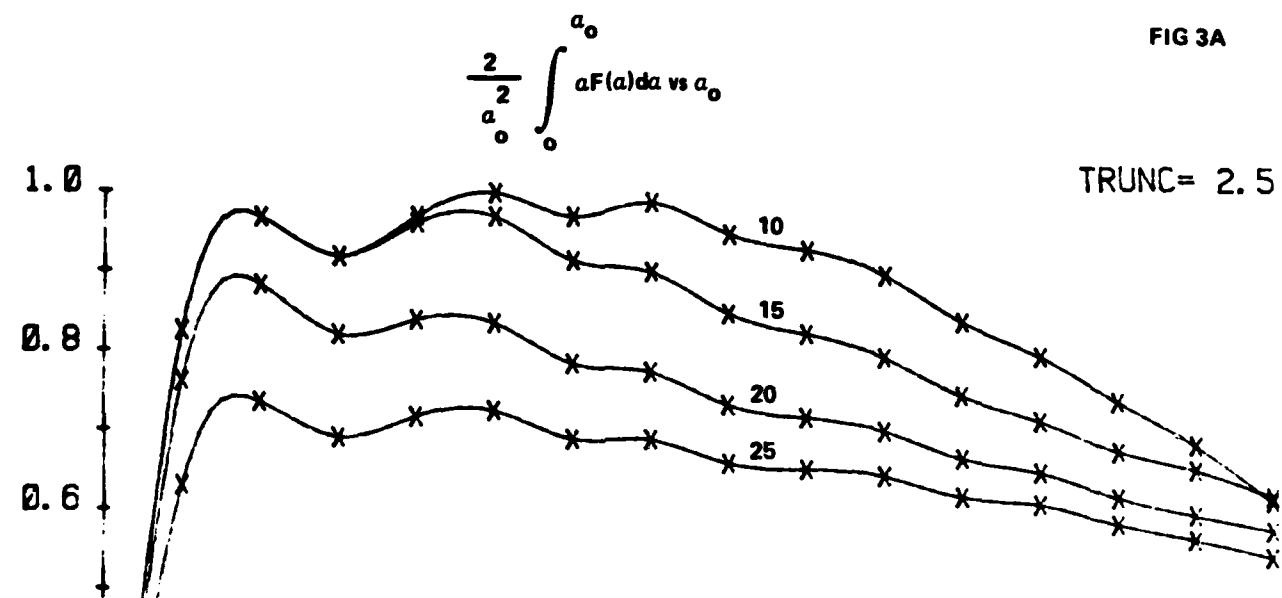


FIG 3B

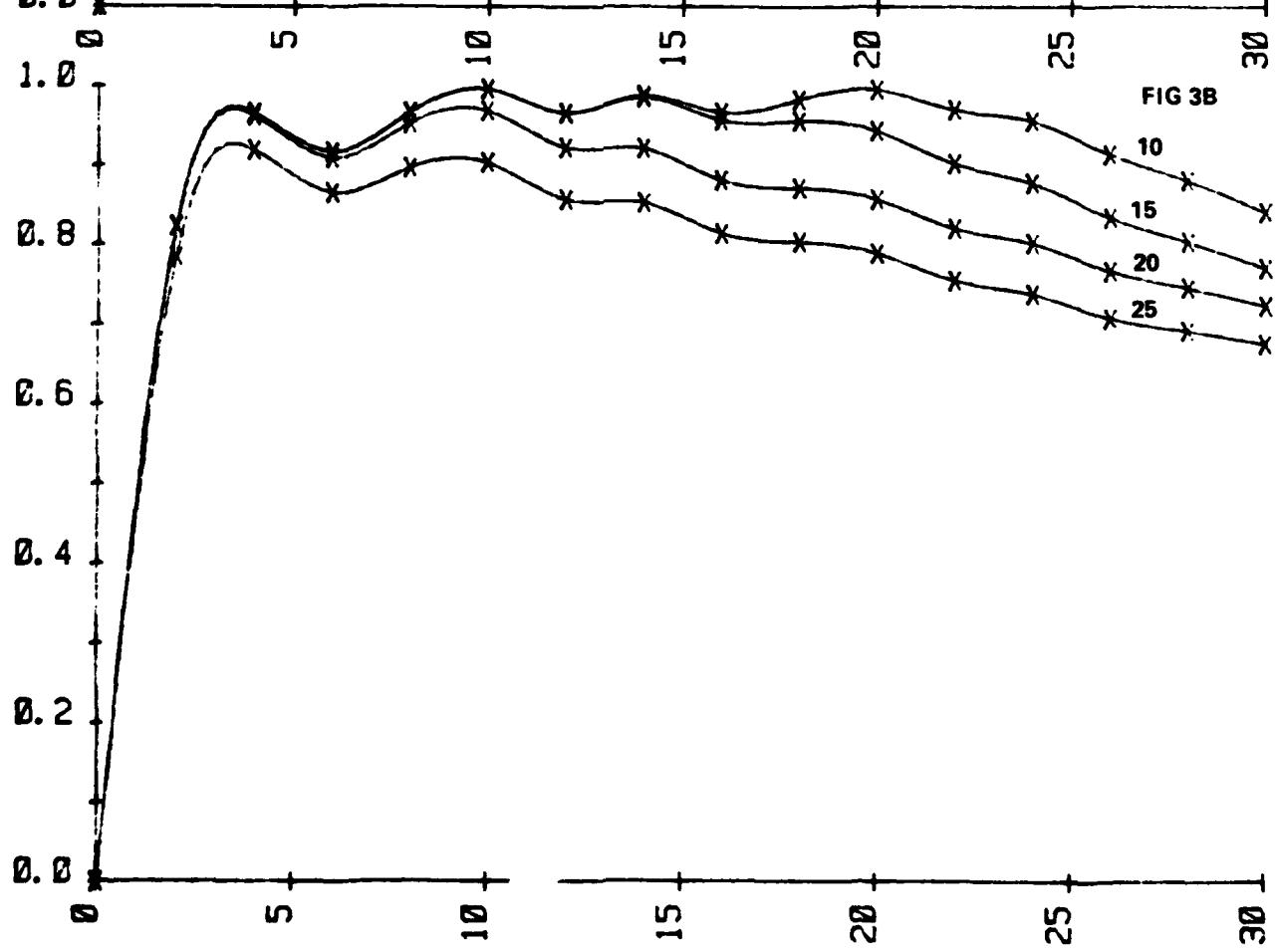


FIG 3C

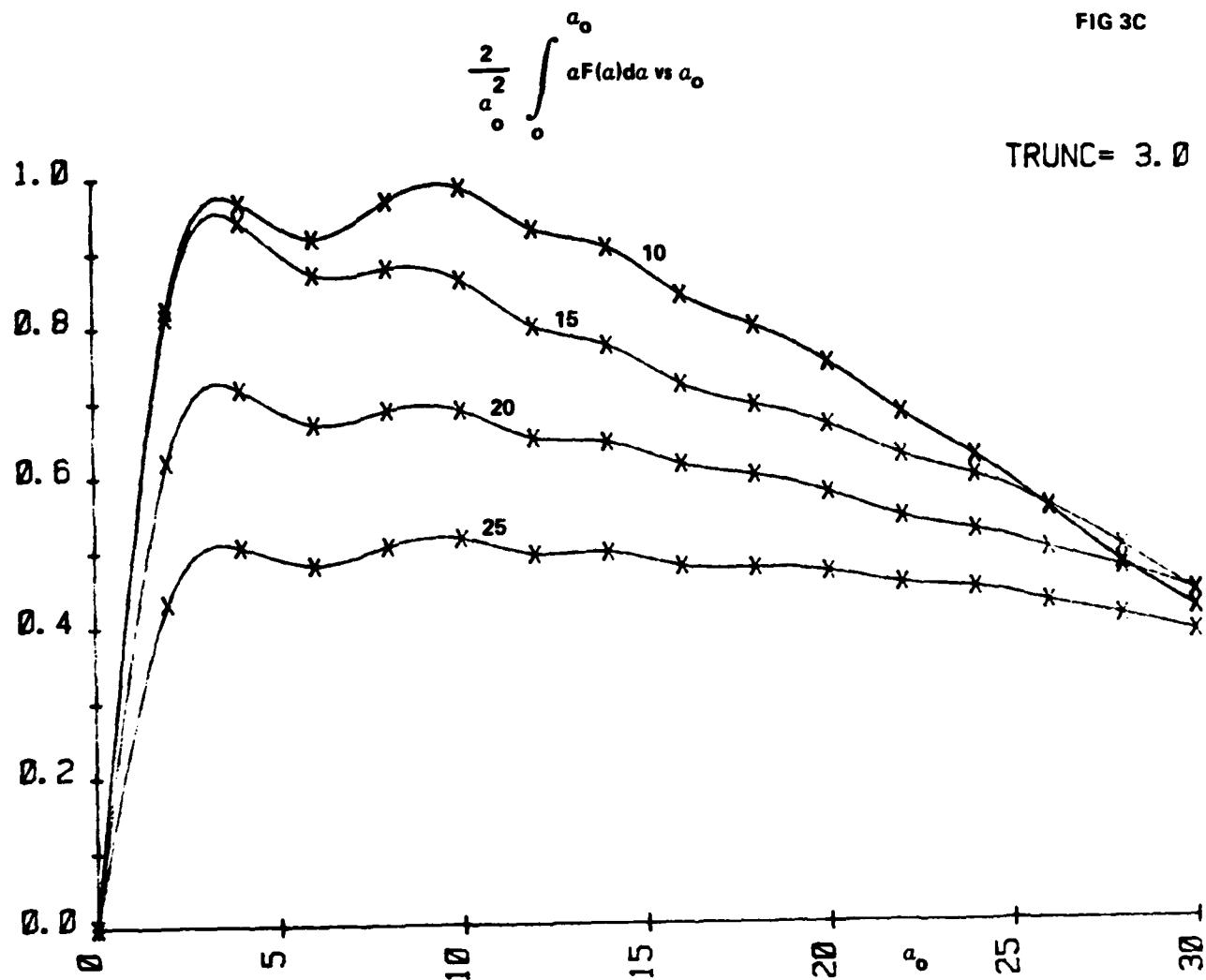


FIG 3D

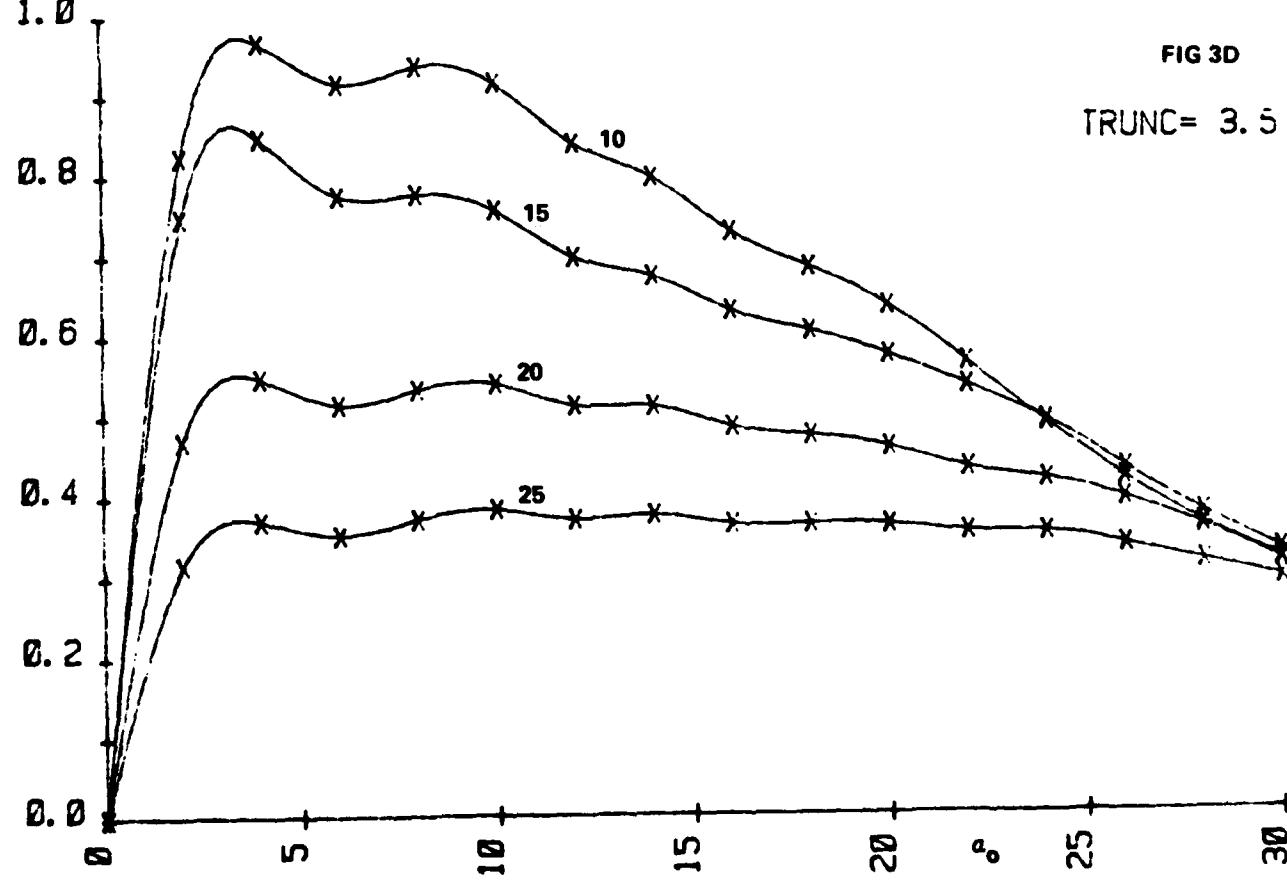


FIG 4A

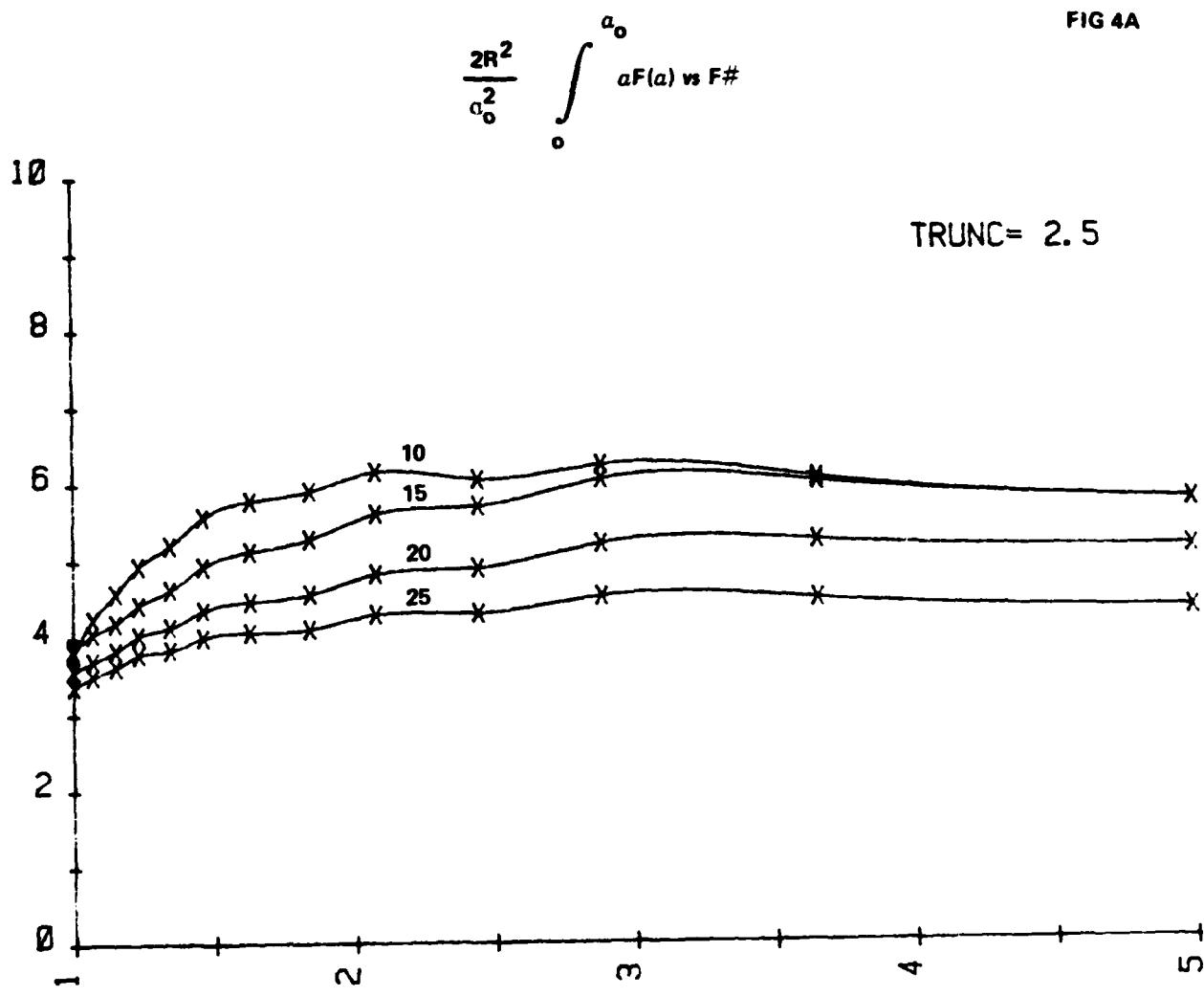


FIG 4B

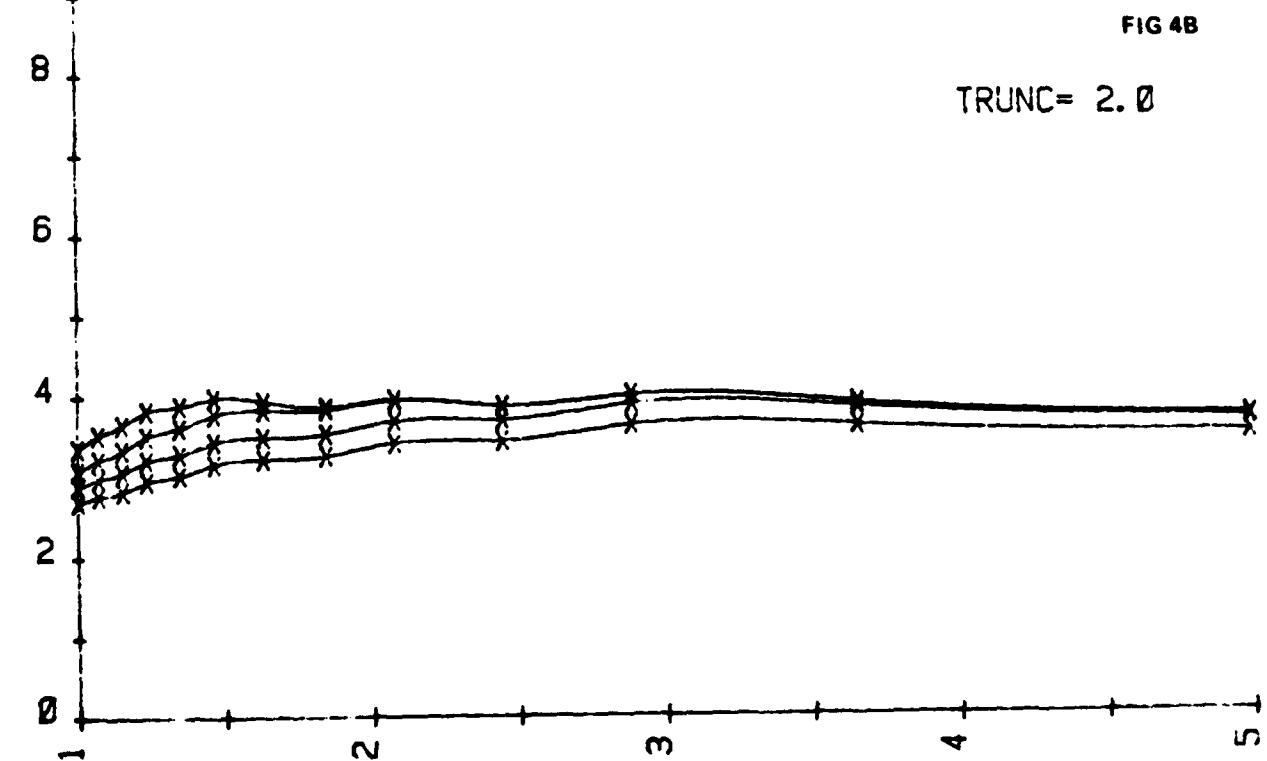


FIG 4C

$$\frac{2R^2}{a_o^2} \int_0^{a_o} aF(a) \, da$$

$$F\# = \frac{1}{2 \sin a_o}$$

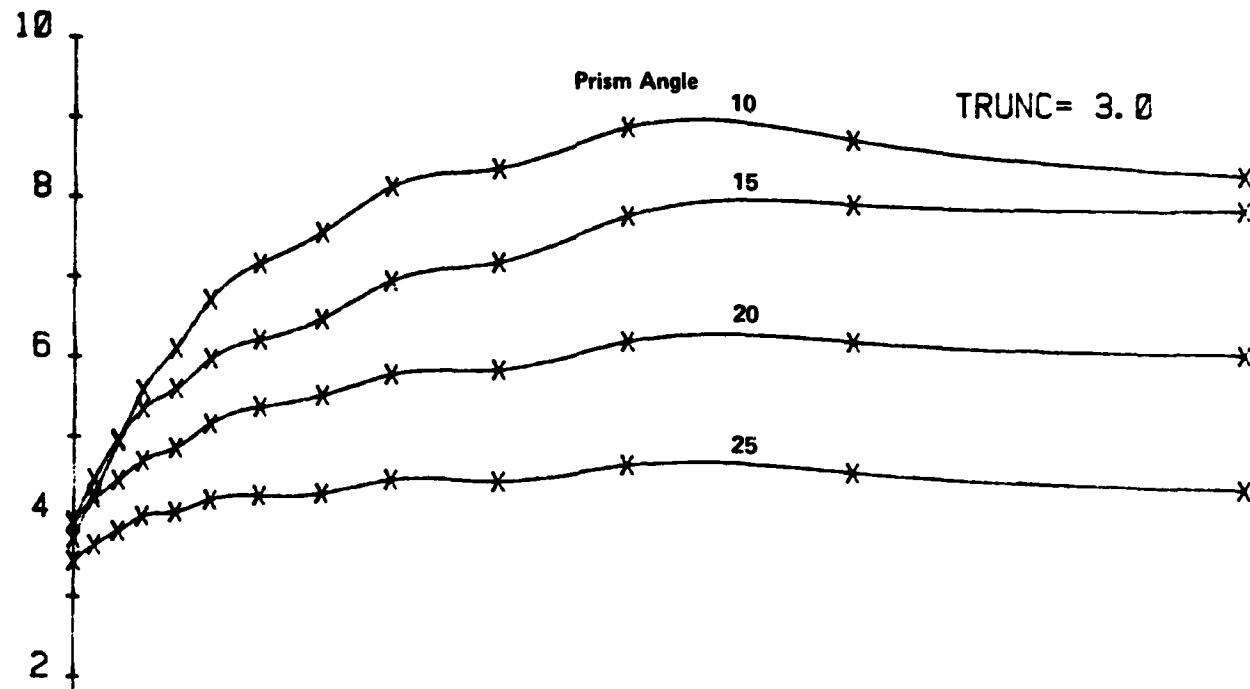
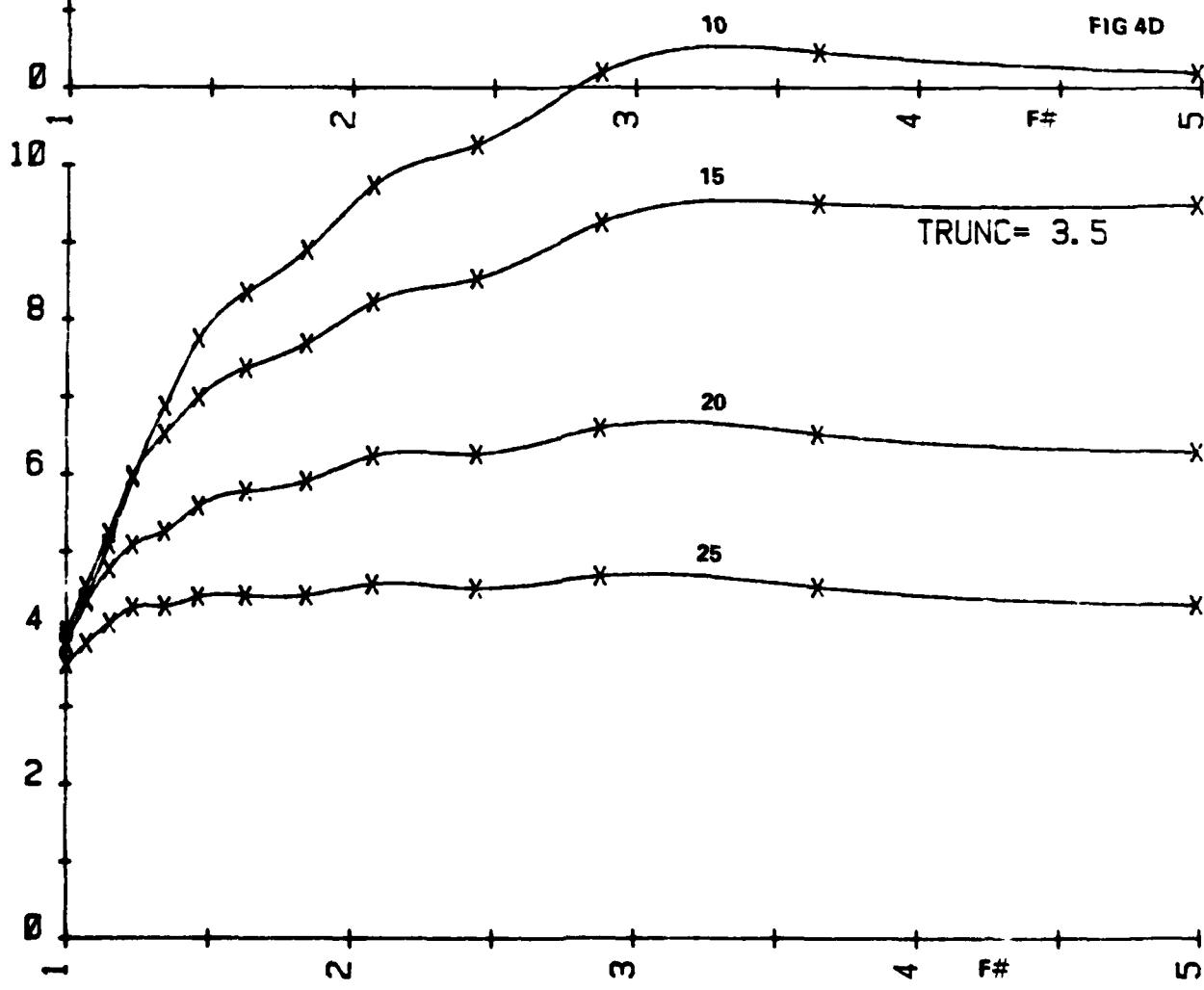


FIG 4D



$$GAIN = \frac{2R^2}{a^2} \int_0^{a_0} a F(a) da \text{ vs } R \text{ for various pyramid height}$$

FIG 5A

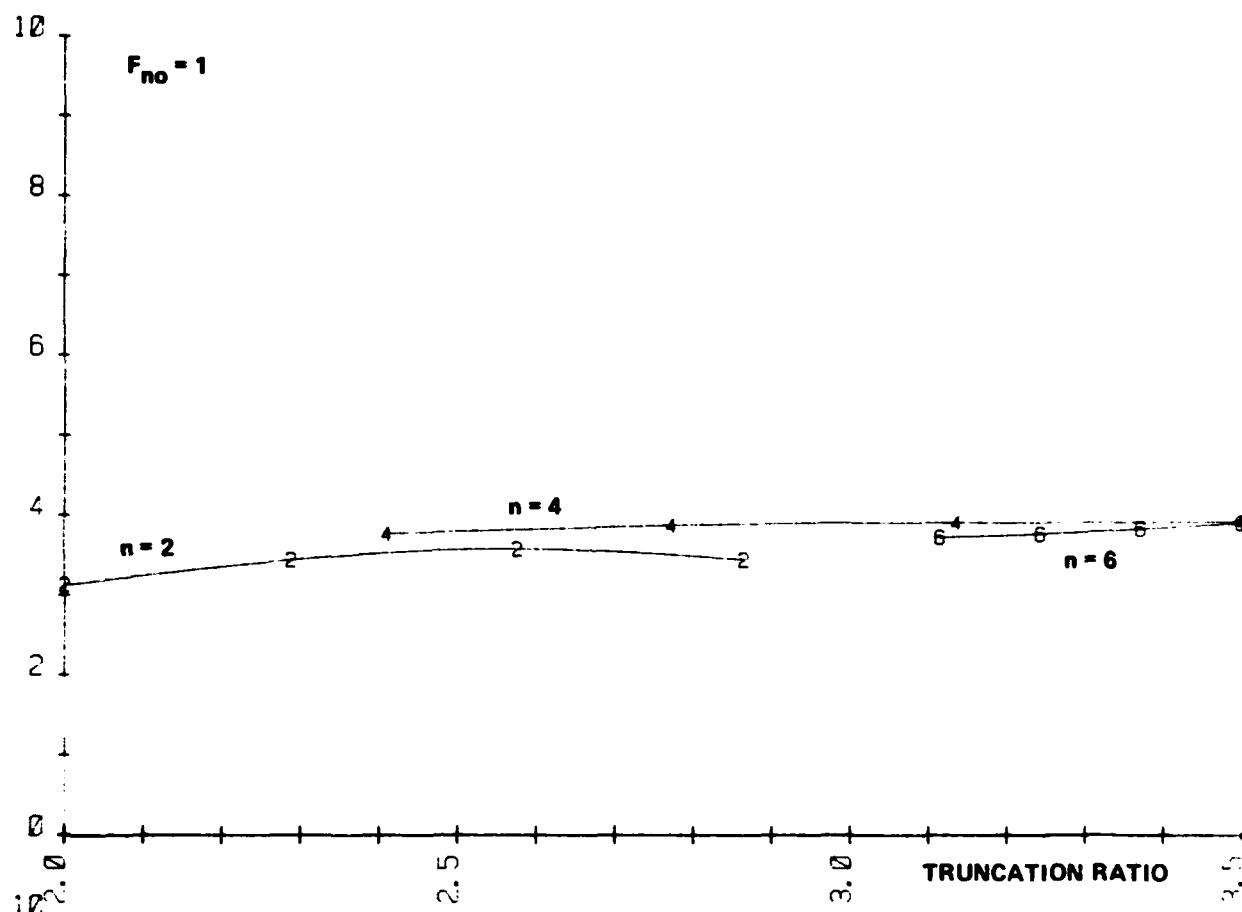


FIG 5B

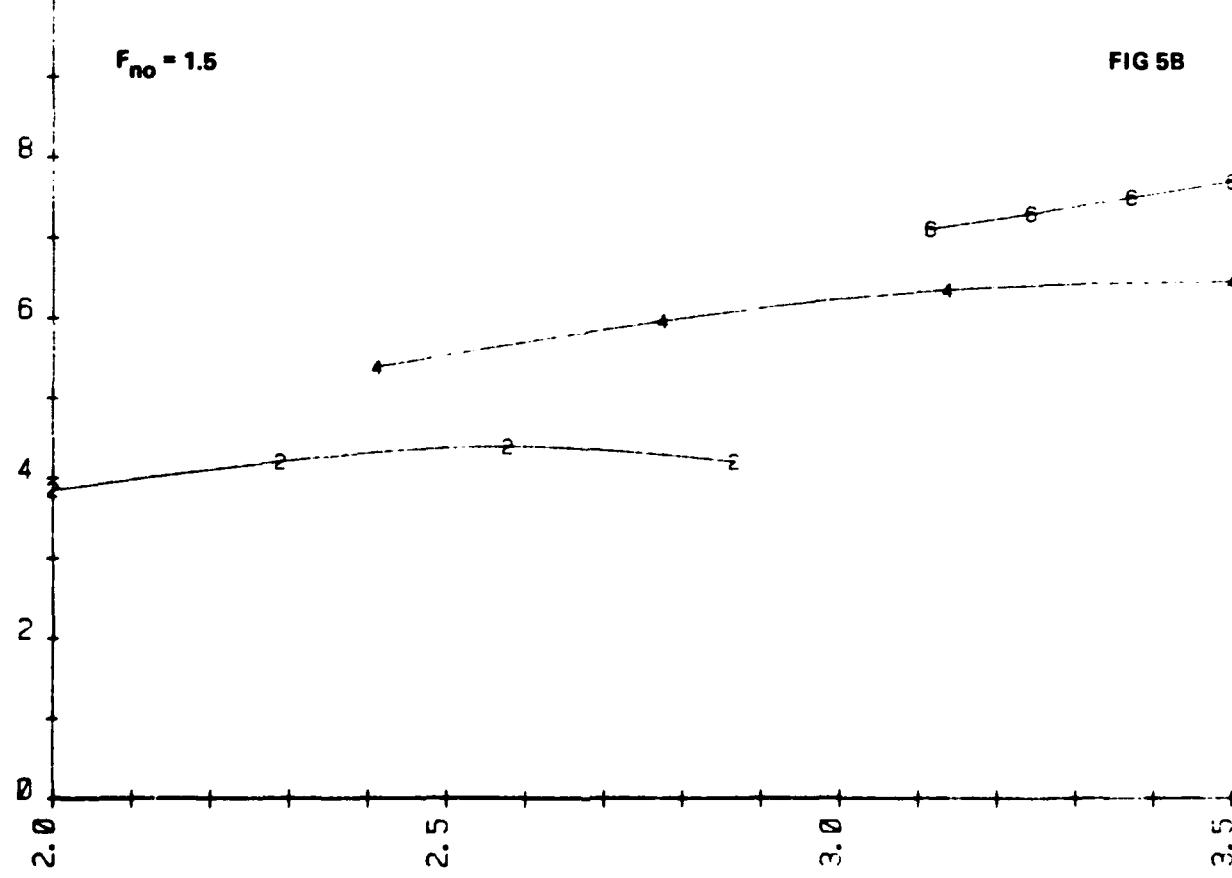


FIG 5C

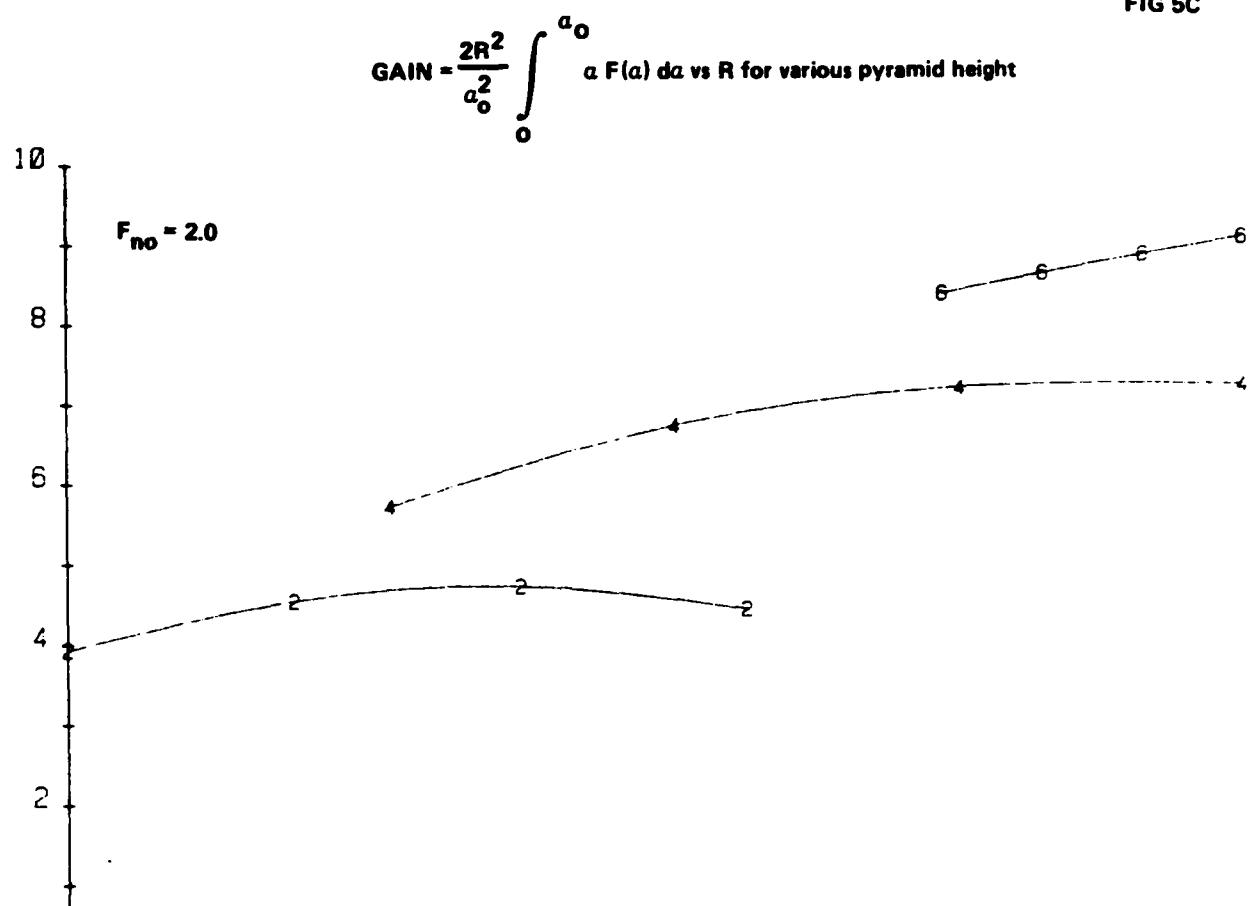
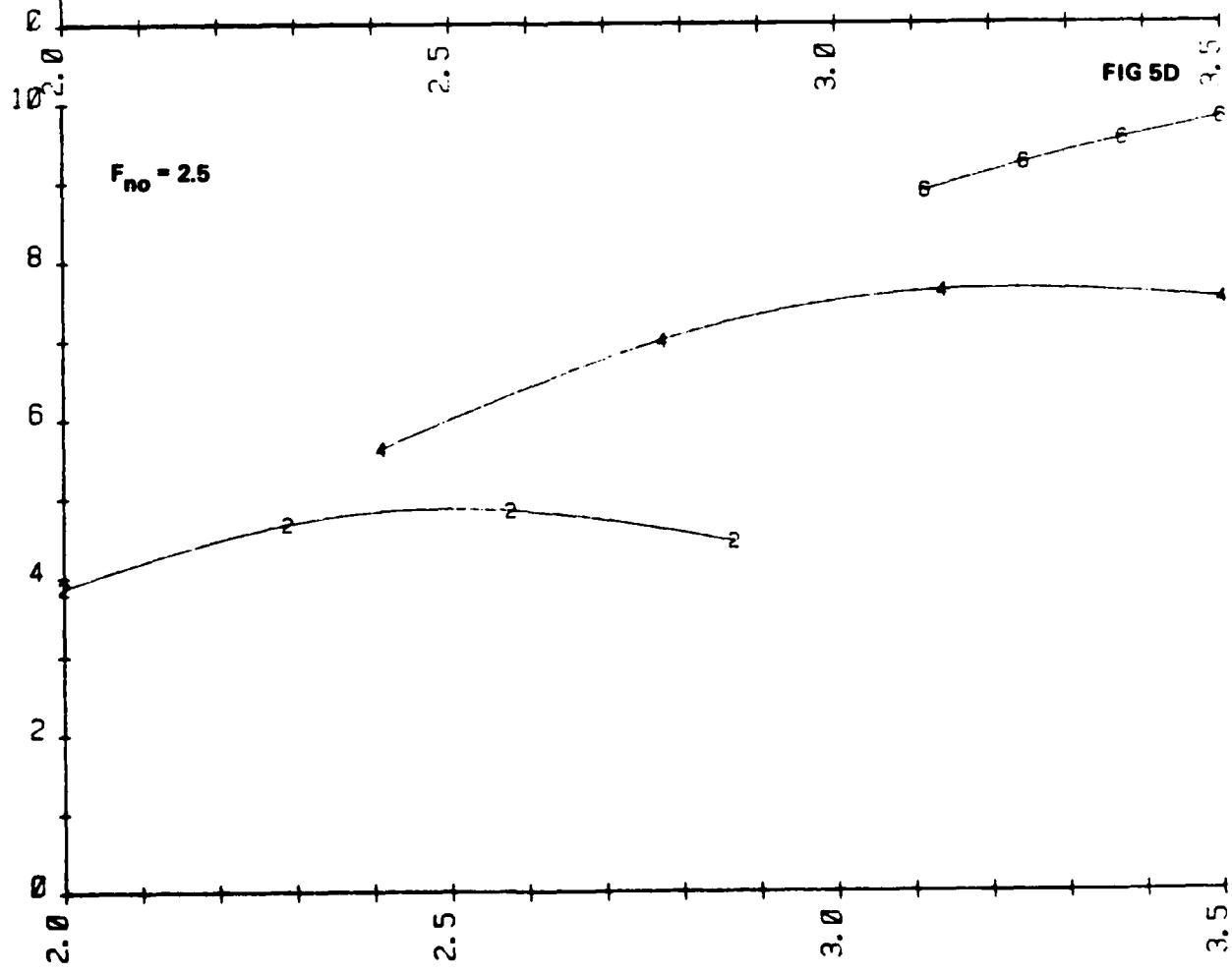


FIG 5D



```

0: "RT2#6":dsp "SPECIAL PYRAMID TRACE (15/12/83)":ato "PROGRAM":stp

61: "PYR":cfg 8,9;ina B;for I=1 to 3;for J=1 to 3;R[p1,I+3J+7]→Z
62: B[I]+Z(A[J]-R[p1,J])→B[I];B[I+3]+ZA[J+3]→B[J+3];next J;next I
63: if B[6]>0;sfq B;ret
64: tan(R[p1,22])→Z;ZB[3]→r3;ZB[6]→r4;sfq 14
65: -(r3+B[1])/(r4+B[4])→r5;if prnd(r5,-8)<=0;9e99→r5
66: -(r3-B[1])/(r4-B[4])→r6;if prnd(r6,-8)<=0;9e99→r6
67: -(r3+B[2])/(r4+B[5])→r7;if prnd(r7,-8)<=0;9e99→r7
68: -(r3-B[2])/(r4-B[5])→r8;if prnd(r8,-8)<=0;9e99→r8
69: cfg 14;min(r5,r6,r7,r8)→r2
70: if r2=9e99;sfq B;ret
71: for I=1 to 3;B[I]+r2B[I+3]→B[I]→r(I+6);next I
72: if abs(B[3])<U/tan(R[1,22]);sfq 9;ret
73: prnd(B[3]+B[1]/Z,-6)→r5
74: prnd(B[3]-B[1]/Z,-6)→r6
75: prnd(B[3]+B[2]/Z,-6)→r7
76: prnd(B[3]-B[2]/Z,-6)→r8
77: (r5=0)+2(r6=0)+3(r7=0)+4(r8=0)→r0
78: (r0=2)-(r0=1)→r7;(r0=4)-(r0=3)→r8
79: -Z→r9;1/f(r7r7+r8r8+r9r9)+Z;Zr7→r7;Zr8→r8;Zr9→r9
80: ina C;0→r10→r11→r12
81: for I=1 to 3;for J=1 to 3;R[p1,7+3I+J]→Z
82: C[I]+ZB[J]→C[I];r(9+I)+Zr(6+J)→r(9+J);next J
83: C[I]+R[p1,I]→C[I];next I;ato "REFLECT";if abs(N)≠9e99;ato "REFRACT"

156: "PROGRAM":
157: if not flag2;sfq 12;dim X[0:20],Y[0:20],F[0:20]
158: ina R;90→R[1,4]+R[1,5]+R[1,8];9e99→R[1,20];-9e99→R[1,9]
159: cll 'SMG'(1)
160: 32→U[1];16→U[2];8→U[3];4→U[4];2→U[5];1→U[6];.5→U[7];.25→U[8]
161: "NEXT":7→Q
162: for W=1 to 5
163: (W=1)+(3/3.5)(W=2)+(3/2.5)(W=3)+(3/2)(W=4)+(3/1.5)(W=5)→U
164: fxd 4;prt "RATIO",3/U;spc
165: for E=10 to 25 by 5;E→R[1,22]
166: prt "P=",E;spc ;0→0
167: for A=0 to 30 by 2;ina F
168: fxd 6;prt "A=",A
169: for P=1 to -1 by -2
170: -cos(A)→D[5];Psin(A)→D[6];3/tan(R[1,22])→D[2]
171: .1→H;for G=0 to 10;.3G→D[1]
172: 0→B→C;64→F;for D=1 to 8;3F/128→D[3]
173: afa D→C;for V=1 to 1000;afa C→A;cll 'RAYTRACE'(1)
174: if flag8;F-U[D]→F;ato +3
175: if flag9;F+U[D]→F;ato +2
176: next V
177: next D
178: D[3]/6+F[G+1]→F[G+1];next G;next P
179: fxd 6;11→N;cll 'SIMP';prt "S=",S;spc
180: A+X[0+1→0];S→Y[0];next A
181: prt "-----"
182: E+X[0];3/U→Y[0];rcf 0+1→Q,X[*],Y[*]
183: next E;next W
184: end

```

```

0: "RT2#30 PYRAMID RESULTS WITH SPLINE FIT (06/01/84)":oto "P"
1: "DERINT":cfa 3;1→I
2: if (I+1→I)>N-1:gto +3
3: .5(X[I]-X[I-1]+X)/(X[I+1]-X[I-1]+H)→R[I]
4: 2(((Y[I+1]-Y[I])/(X[I+1]-X[I])-(Y[I]-Y[I-1])/X)/H+T)→S[I];3T→G[I]:oto -2
5: 0→S[1]→S[N];B-4/3→W
6: 0→U;2→I
7: W(-S[I]-R[I]S[I-1]-(.5-B[I])S[I+1]+G[I])→T
8: if (abs(T)+H)>U;H→U
9: S[I]+T→S[I]
10: if I#N-1;I+1→I:gto -3
11: if U>=E:gto -5
12: 0→I
13: if (I+1→I)>N-1:gto +2
14: (S[I+1]-S[I])/(X[I+1]-X[I])→G[I]:oto -1
15: if M=0:gto +13
16: 0→J
17: if (J+1→J)>M:gto +11
18: 1→I;if (T[J]+T)>=X[I];gto +2
19: sfa 3;dsp "ARG OUT OF BOUND";ret
20: if (I+1→I)>N:gto -1
21: if T>X[I];gto -1
22: I-1→I
23: T[I]-X[I]→H;T[J]-X[I+1]→T;HT→X
24: S[I]+HG[I]→S
25: (1/6+Z)(S[I]+S[I+1]+S)→U
26: ((Y[I+1]-Y[I])/(X[I+1]-X[I])→W)H+Y[I]+XU+R[J]
27: W+(H+T)U+ZXG[I]→D[J];gto -10
28: 0→I→A
29: if (I+1→I)>N-1:ret
30: A+(.5(X[I+1]-X[I]+H)(Y[I]+Y[I+1])-(1/24)H↑3(S[I]+S[I+1]))→A:gto -1
31:
32: "P":
33: 21→N;dim X[N],Y[N],S[N],G[N-1],U[21],V[21]
34: 100→M;dim T[M],B[M],D[M]
35: fmt ;wrt 705,"IP700,800,7100,10400"
36: wrt 705,"VS15";csiz 1.3,1.7.1.5
37: 7→0
38: "NEXT":for Y=1 to 2
39: 1.1+Z;sc1 0,30,-Z(2-Y),ZY
40: fxd 0;xax 0.5,0,30,1;fxd 1;yax 0..1,0,1,2;fxd 6
41: for C=1 to 4
42: ldf 0+1+0,U[*],V[*]
43: if C=1;fxd 1;plt 25,.95,1;1b1 "TRUNC=",V[1];fxd 6
44: for I=1 to 20;U[I+1]→X[I];V[I+1]→Y[I];next I
45: csiz 1.1.6,3/2;wrt 705,"SMX";for I=1 to 16;plt X[I],Y[I],1;next I
46: wrt 705,"SM"
47: for I=0 to 99;30I/99→T[I+1];next I
48: 16→N;100→M;1e-6→E;gsb "DERINT"
49: for I=1 to 100;plt T[I],B[I];next I;pen
50: next C
51: next Y
52: end

```

```

0: "RT2#31 PYRAMID RESULTS WITH SPLINE FIT (ii) (09/01/84)":ato "P"
1: "DERINT":cfg 3:I+I
2: if (I+1+I)>N-1:ato +3
3: .5(X[I]-X[I-1]+X)/(X[I+1]-X[I-1]+H)+B[I]
4: 2(((Y[I]+1)-Y[I]))/(X[I+1]-X[I])-((Y[I]-Y[I-1])/X)/H+T)+S[I];3T+G[I];ato -2
5: 0+S[I]+S[N];B-4F3+W
6: 0+U;2+I
7: W(-S[I]-B[I]S[I-1]-(.5-B[I])S[I+1]+G[I])+T
8: if (abs(T)+H)>U;H+U
9: S[I]+T+S[I]
10: if I>N-1;I+1+I;ato -3
11: if U>E;ato -5
12: 0+I
13: if (I+1+I)>N-1;ato +2
14: (S[I+1]-S[I])/(X[I+1]-X[I])+G[I];ato -1
15: if M=0;ato +13
16: 0+J
17: if (J+1+J)>M;ato +11
18: 1+I;if (T[J]+T)=X[I];ato +2
19: sfq 3;dsp "ARG OUT OF BOUND";ret
20: if (I+1+I)>N;ato -1
21: if T>X[I];ato -1
22: I-1+I
23: T[J]-X[I]+H;T[J]-X[I+1]+T;HT+X
24: S[I]+HG[I]+S
25: (1/6+Z)(S[I]+S[I+1]+S)+U
26: ((Y[I+1]-Y[I]))/(X[I+1]-X[I])+W)H+Y[I]+XU+B[J]
27: W+(H+T)U+ZXG[I]+D[J];ato -10
28: 0+I+A
29: if (I+1+I)>N-1;ret
30: A+(.5(X[I+1]-X[I]+H)(Y[I]+Y[I+1]))-(1/24)H^3(S[I]+S[I+1]))+A;ato -1
31:
32: "P";
33: 21+N;dim X[N],Y[N],S[N],G[N-1],U[21],V[21]
34: 100+M;dim T[M],B[M],D[M]
35: fmt :wrt 705,"IP700,800,7100,10400"
36: wrt 705,"VS15";csiz 1.3,1.7,1.5
37: 7+Q
38: "NEXT":for Y=1 to 2;15+B
39: 1.1+Z;scl 0,30,-ZB(2-Y),ZBY
40: fxd 0;xax 0,5,0,30,1;fxd 0;yax 0,1,0,15,5;fxd 6
41: for C=1 to 4
42: ldf 0+1+Q,U[*],V[*]
43: if C=1;fxd 1;plt 25,B,1;1b] "TRUNC=",V[1];fxd 6
44: for I=1 to 20;U[I+1]+X[I];V[I+1]X[I]+Y[I];next I
45: csiz 1,1.6,3/2;wrt 705,"SMX";for I=1 to 16;plt X[I],Y[I],1;next I
46: wrt 705,"SM"
47: for I=0 to 99;30I/99+T[I+1];next I
48: 16+N;100+M;1e-6+E;gsb "DERINT"
49: for I=1 to 100;plt T[I],B[I];next I;pen
50: next C
51: next Y
52: end

```

```

0: "RT2#32 PYRAMID RESULTS WITH SPLINE FIT (iii) (09/01/84)":oto "P"
1: "DERINT":cfq 3;1+I
2: if (I+1+I)>N-1:gto +3
3: .5(X[I]-X[I-1]+X)/(X[I+1]-X[I-1]+H)+B[I]
4: 2(((Y[I+1]-Y[I])/(X[I+1]-X[I])-(Y[I]-Y[I-1])/X)/H+T)+S[I]:3T+G[I]:oto -2
5: 0+S[I]+S[N];B-4F3+T
6: 0+U:2+I
7: W(-S[I]-B[I]S[I-1]-(.5-B[I])S[I+1]+G[I])+T
8: if (abs(T)+H)>U;H+U
9: S[I]+T+S[I]
10: if I#N-1;I+1+I:gto -3
11: if U>E:gto -5
12: 0+I
13: if (I+1+I)>N-1:gto +2
14: (S[I+1]-S[I])/(X[I+1]-X[I])+G[I]:oto -1
15: if M=0:gto +13
16: 0+J
17: if (J+1+J)>M:gto +11
18: 1+I;if (T[J]+T)=X[I];oto +2
19: sfq 3;dsp "ARG OUT OF BOUND";ret
20: if (I+1+I)>N:gto -1
21: if T>X[I];gto -1
22: I-1+I
23: T[J]-X[I]+H;T[J]-X[I+1]+T;HT+X
24: S[I]+HG[I]+S
25: (1/6+Z)(S[I]+S[I+1]+S)+U
26: ((Y[I+1]-Y[I])/(X[I+1]-X[I])+W)H+Y[I]+XU+B[J]
27: W+(H+T)U+ZXG[I]+D[J]:gto -10
28: 0+I+A
29: if (I+1+I)>N-1;ret
30: A+(.5(X[I+1]-X[I]+H)(Y[I]+Y[I+1])-(1/24)H+3(S[I]+S[I+1]))+A:oto -1
31:
32: "P":
33: 100→N;dim X[N],Y[N],S[N],G[N-1],U[21],V[21]
34: 100→M;dim T[M],B[M],D[M]
35: fpt ;wrt 705,"IP700,800,7100,10400"
36: wrt 705,"VS15";csiz 1.3,1.7,1.5
37: 7+Q
38: "NEXT":for Y=1 to 2;1+Y
39: 1.1+Z;scl 0,30,-ZB(2-Y),ZBY
40: csiz 1.3,1.7,1.5
41: fxd 0;xax 0,5,0,30,1;fxd 1;yax 0,.1,0,1,2;fxd 6
42: for C=1 to 4
43: ldf 0+1+Q,U[*],V[*];V[1]+R;U[1]+P
44: if C=1;fxd 1;plt 25,B,1:lbl "TRUNC=",V[1]:fxd 6
45: for I=1 to 20;U[I+1]+X[I];V[I+1]X[I]+Y[I];next I
46: dsp "recalculating 100 X[*],Y[*]"
47: for I=0 to 99;30I/99+T[I+1];next I
48: 100→M;16→N;1e-6→E;gsb "DERINT"
49: rdm X[M],Y[M];ara T+X;ara B+Y;dsp
50: rdm U[N],V[N]
51: 0+U[1]+V[1];1+F;for D=2 to 30 by 2;0+M;F+1+F
52: for I=1 to M;(I-1)D/(M-1)+T[I]:next I
53: int(3.3D)+1→N;1e-6→E;gsb "DERINT"
54: dsp D,2A/D+2+Z:D+U[F];Z+V[F];next D
55: rdm X[16],Y[16];ara U+X;ara V+Y
56: csiz 1,1.6,3/2;wrt 705,"SMX";for I=1 to 16;plt X[I],Y[I],1:next I
57: wrt 705,"SM"
58: for I=0 to 99;30I/99+T[I+1];next I
59: 16+M;100→M;1e-6→E;gsb "DERINT"
60: for I=1 to 100;plt T[I],B[I];next I;pen
61: next C
62: next Y
63: end

```

```

0: "RT2#33 PYRAMID RESULTS WITH SPLINE FIT (iv) (09/01/84)":ato "P"
1: "DERINT":cfa 3:1+I
2: if (I+1+I)>N-1:ato +3
3: .5(X[I]-X[I-1]+X)/(X[I+1]-X[I-1]+H)+R[I]
4: 2*((Y[I+1]-Y[I])/(X[I+1]-X[I])-(Y[I]-Y[I-1])/X)/H+T)+S[I];3T+G[I]:ato -2
5: 0+S[I]+S[N];B-4T3+W
6: 0+U:2+I
7: W(-S[I]-B[I]S[I-1]-(.5-B[I])S[I+1]+G[I])+T
8: if (abs(T)+H)>U: H+U
9: S[I]+T+T+S[I]
10: if I#N-1;I+1+I:qto -3
11: if U=E:qto -5
12: 0+I
13: if (I+1+I)>N-1:ato +2
14: (S[I+1]-S[I])/(X[I+1]-X[I])+G[I];qto -1
15: if M=0:qto +13
16: 0+J
17: if (J+1+J)>M:qto +11
18: 1+I;if (T[J]+T)=X[I]:qto +2
19: sfq 3;dsp "ARG OUT OF BOUND";ret
20: if (I+1+I)>N:qto -1
21: if T>X[I]:qto -1
22: I-1+I
23: T[J]-X[I]+H;T[J]-X[I+1]+T;HT+X
24: S[I]+HG[I]+S
25: (1/6+Z)(S[I]+S[I+1]+S)+U
26: ((Y[I+1]-Y[I])/(X[I+1]-X[I])+W)H+Y[I]+XU+R[J]
27: W+(H+T)U+ZXG[I]+D[J];qto -10
28: 0+I+A
29: if (I+1+I)>N-1;ret
30: A+(.5(X[I+1]-X[I]+H)(Y[I]+Y[I+1])-(1/24)H+3(S[I]+S[I+1]))+A;ato -1
31:
32: "P":
33: 100+N;dim X[N],Y[N],S[N],G[N-1],U[21],V[21]
34: 100+M;dim T[M],B[M],D[M]
35: fmt ;wrt 705,"IP700,800,7100,10400"
36: wrt 705,"VS15";csiz 1.3,1.7,1.5
37: 7+Q
38: "NEXT":for Y=1 to 2;10+B
39: 1.1+Z;scl 1.5,-ZB(2-Y),ZBY
40: csiz 1.3,1.7,1.5
41: fxd 0;xax 0,.5,1,5,2;fxd 0;yax 1,1,0,10,2;fxd 6
42: for C=1 to 4;rdm U[21],V[21],X[20],Y[20]
43: ldf 0+1+0,U[*],V[*];V[1]+R;U[1]+P
44: if C=1;fxd 1;plt 4,9,1;lbl "TRUNC=",V[1];fxd 6
45: for I=1 to 20;U[I+1]+X[I];V[I+1]+Y[I];next I
46: dsp "recalculating 100 X[*],Y[*]"
47: for I=0 to 99;30I/99+T[I+1];next I
48: 100+M;16+N;1e-6+E;gsb "DERINT"
49: rdm X[M],Y[M];ara T+X;ara B+Y;dsp
50: rdm U[N],V[N]
51: 0+U[1]+V[1];1+F;for D=2 to 30 by 2;0+M;F+1+F
52: for I=1 to M;(I-1)D/(M-1)+T[I];next I
53: int(3.3D)+1+N;1e-6+E;gsb "DERINT"
54: dsp D,2ARR/D+2+Z;X[N]+U[F];Z+V[F];next D
55: rdm X[16],Y[16];for I=1 to 13;V[17-I]+Y[I];1/2sin(U[17-I])+X[I];next I
56: csiz 1,1.6,3/2;wrt 705,"SMX";for I=1 to 13;plt X[I],Y[I],1;next I
57: wrt 705,"SM"
58: for I=0 to 99;1+(X[13]-1)I/99+T[I+1];next I
59: 13+N;100+M;1e-6+E;gsb "DERINT"
60: for I=1 to 100;plt T[I],B[I];next I;pen
61: next C
62: next Y
63: end

```

```

0: "RT2#34 PYRAMID RESULTS WITH SPLINE FIT (v) (12/01/84)":oto "P"
1: "DERINT":cfa 3;1+I
2: if (I+1+I)>N-1:gto +3
3: .5(X[I]-X[I-1]+X)/(X[I+1]-X[I-1]+H)+R[I]
4: 2(((Y[I+1]-Y[I])/(X[I+1]-X[I])-(Y[I]-Y[I-1])/X)/H+T)+S[I]:3T+G[I];gto -2
5: 0+S[I]+S[N];8-4f3+W
6: 0+U;2+I
7: W(-S[I]-B[I]S[I-1]-(.5-B[I])S[I+1]+G[I])+T
8: if (abs(T)+H)>U;H+U
9: S[I]+T+S[I]
10: if I+N-1;I+1+I:gto -3
11: if U=E:gto -5
12: 0+I
13: if (I+1+I)>N-1:gto +2
14: (S[I+1]-S[I])/(X[I+1]-X[I])+G[I];gto -1
15: if M=0:gto +13
16: 0+J
17: if (J+1+J)>M:gto +11
18: 1+I;if (T[J]+T)>=X[1];gto +2
19: sfp 3;dsp "ARG OUT OF BOUND";ret
20: if (I+1+I)>N:gto -1
21: if T>X[I];gto -1
22: I-1+I
23: T[J]-X[I]+H;T[J]-X[I+1]+T;HT+X
24: S[I]+HG[I]+S
25: (1/6+Z)(S[I]+S[I+1]+S)+U
26: ((Y[I+1]-Y[I])/(X[I+1]-X[I])+W)H+Y[I]+XU+B[J]
27: W+(H+T)U+ZXG[I]+D[J];gto -10
28: 0+I+A
29: if (I+1+I)>N-1:ret
30: A+(.5(X[I+1]-X[I]+H)(Y[I]+Y[I+1])-(1/24)H↑3(S[I]+S[I+1]))+A;gto -1
31:
32: "P";
33: 100→N;dim X[N],Y[N],S[N],G[N-1],U[21],V[21]
34: 100→M;dim T[M],B[M],D[M]
35: dim R[4,4,4]
36: "NEXT":for Q=8 to 8+16
37: rdm U[21],V[21],X[20],Y[20]
38: ldf Q,U[*],V[*];V[1]→R:U[1]+P
39: for I=1 to 20;U[I+1]+X[I];V[I+1]X[I]+Y[I];next I
40: dsp "recalculating 100 X[*],Y[*]"
41: for I=0 to 99;30I/99→T[I+1];next I
42: 100→M;16→N;1e-6→E;gsb "DERINT"
43: rdm X[M],Y[M];ara T→X;ara B→Y;dsp
44: rdm U[N],V[N]
45: 0+U[1]+V[1];1+F;for D=2 to 30 by 2;0+M;F+1+F
46: for I=1 to M;(I-1)D/(M-1)+T[I]:next I
47: int(3.3D)+1+N;1e-6→E;gsb "DERINT"
48: dsp D,2ARR/D↑2+Z;X[N]+U[F];Z+V[F]:next D
49: rdm X[16],Y[16];for I=1 to 13;V[17-I]+Y[I];1/2sin(U[17-I])+X[I];next I
50: for I=1 to 4;1+.5(I-1)+T[I]:next I
51: 13+N;4+M;1e-6→E;gsb "DERINT"
52: for I=1 to 4;B[I]+R[I,2R-3,.2P-1]:next I
53: next Q
54: end

```

```

0: "RT2#36 PYRAMID RESULTS WITH SPLINE FIT (vj) (27/01/84)":ato "P"
1: "DERINT":cfa 3;1+I
2: if (I+1+I)>N-1:ato +3
3: .5(X[I]-X[I-1]+X)/(X[I+1]-X[I-1]+H)+B[I]
4: 2(((Y[I+1]-Y[I])/(X[I+1]-X[I])-(Y[I]-Y[I-1])/X)/H+T)+S[I]:3T+G[I];ato -2
5: 0+S[I]+S[N];B-4/3+W
6: 0+U:2+I
7: W(-S[I]-B[I]S[I-1]-(.5-B[I])S[I+1]+G[I])-T
8: if (abs(T)+H)>U:H+U
9: S[I]+T+S[I]
10: if I#N-1;I+1+I:ato -3
11: if U=E:qto -5
12: 0+I
13: if (I+1+I)>N-1:qto +2
14: (S[I+1]-S[I])/(X[I+1]-X[I])+G[I]:ato -1
15: if M=0:qto +13
16: 0+J
17: if (J+1+J)>M:qto +11
18: 1+I;if (T[J]+T)>=X[I];ato +2
19: sfq 3;dsp "ARG OUT OF BOUND";ret
20: if (I+1+I)>N:qto -1
21: if T>X[I]:qto -1
22: I-1+I
23: T[J]-X[I]+H;T[J]-X[I+1]+T;HT+X
24: S[I]+HG[I]+S
25: (1/6+Z)(S[I]+S[I+1]+S)+U
26: ((Y[I+1]-Y[I])/(X[I+1]-X[I])+W)H+Y[I]+XU+B[J]
27: W+(H+T)U+ZXG[I]+D[J];qto -10
28: 0+I+I
29: if (I+1+I)>N-1:ret
30: A+(.5(X[I+1]-X[I]+H)(Y[I]+Y[I+1])-(1/24)H^3(S[I]+S[I+1]))+A:qto -1
31: "P":
32: 100→N;dim X[N],Y[N],S[N],G[N-1],U[21],V[21]
33: 100→M;dim T[M],B[M],D[M]
34: dim R[4,4,4],Q[4,4]
35: 1df 35,R[*]
36: fmt ;wrt 705,"IP700,800,7100,10400"
37: wrt 705,"VS5";csiz 1.3,1.7,1.5
38: 0+Q
39: "NEXT":for Y=1 to 2;10→B;0+1+Q
40: 1.1→Z;sc1 2,3.5,-ZB(2-Y).ZBY
41: csiz 1.1,1.7,1.5
42: fxd 1;xax 0,.1,2,3.5,5;fxd 0;yax 2,1,0,10,2;fxd 6
43: for L=2 to 6 by 2
44: max(2,2Ltan(10)+1)+r1+T[1]
45: min(3.5,2Ltan(25)+1)+r2+T[4]
46: r1+(r2-r1)/3+T[2];r1+2(r2-r1)/3+T[3]
47: for C=1 to 4
48: for I=1 to 4;R[Q,I,C]+Y[I];.5(3+I)+X[I];next I
49: 4+M→N;1e-6→E;qsb "DERINT"
50: for I=1 to 4;B[I]+Q[C,I];next I;next C
51: for C=1 to 4
52: for I=1 to 4;B[I,C]+Y[I];5(I+1)+X[I];next I
53: atn(.5(T[C]-1)/L)+P+T[1]
54: 1+M;4+N;1e-6→E;qsb "DERINT"
55: B[1]+Q[1,C];next C;r1+T[1]
56: for I=1 to 4;B[1,I]+Y[I];T[I]+X[I];next I
57: csiz .8,1.6,3/2
58: if L=2:wrt 705,"SM2"
59: if L=4:wrt 705,"SM4"
60: if L=6:wrt 705,"SM6"
61: for I=1 to 4;plt X[I],Y[I],1;next I;wrt 705,"SM"
62: for I=0 to 49;r1+(r2-r1)I/49+T[I+1];next I
63: 50+M;4+N;1e-6→E;qsb "DERINT"
64: for I=1 to 50;plt T[I],B[I];next I;pen
65:
66: next L;next Y;end

```

## DOCUMENT CONTROL SHEET

Overall security classification of sheet ... **UNCLASSIFIED** .....

(As far as possible this sheet should contain only unclassified information. If it is necessary to enter classified information, the box concerned must be marked to indicate the classification eg (R) (C) or (S) )

1. DRIC Reference (if known)	2. Originator's Reference MEMORANDUM 3661	3. Agency Reference	4. Report Security U/C Classification
5. Originator's Code (if known)	6. Originator (Corporate Author) Name and Location ROYAL SIGNALS AND RADAR ESTABLISHMENT		
5a. Sponsoring Agency's Code (if known)	6a. Sponsoring Agency (Contract Authority) Name and Location		
7. Title <b>PERFORMANCE ESTIMATE OF A REFLECTING LIGHT PIPE FOR INFRARED DETECTORS</b>			
7a. Title in Foreign Language (in the case of translations)			
7b. Presented at (for conference papers) Title, place and date of conference			
8. Author 1 Surname, initials WARNER, J	9(a) Author 2	9(b) Authors 3,4...	10. Date
11. Contract Number	12. Period	13. Project	14. Other Reference
15. Distribution statement <b>UNLIMITED</b>			
Descriptors (or keywords)			
continue on separate piece of paper			
Abstract			
<p>A hollow reflecting cone or pyramid can be used to concentrate infrared radiation on to a small photo-detector. This note describes calculations which predict that a five fold increase in flux density should be readily obtainable.</p>			

SEARCHED

FILMED

INDEXED

FILED

SEARCHED

FILMED

INDEXED

FILED